GTSuite

Metadata-Driven Application Development Platform

Introduction & System Overview

"Design once in metadata, deploy everywhere: web, iOS, Android.
Changes without recompilation.
Evolution without migration."

GTsoftware di Giancarlo Thiella

Website: www.gtsoftware.ch

Email: giancarlo.thiella@gtsoftware.ch

Version 1.0 - November 2025

Table of Contents

1. What is GTSuite?

GTSuite is a comprehensive platform for building metadata-driven enterprise applications. Rather than hard-coding user interfaces and business logic, GTSuite applications are defined through metadata - structured data that describes what the application should do and how it should appear.

The Core Idea

Imagine building an application where the user interface, forms, grids, menus, validation rules, and business logic are all stored as data in a database rather than written in code. When you need to add a new field, change a form layout, or modify business rules, you update the metadata instead of rewriting and redeploying code.

Aspect	Traditional Approach	GTSuite Approach
UI Changes	Modify code, recompile, redeploy	Update metadata, refresh browser
New Feature	Write code, test, deploy	Design in visual tool, upload metadata
Form Design	HTML templates in code	Visual designer, stored as data
Business Logic	Hardcoded in application	Defined as metadata actions
Multi-Platform	Separate codebase per platform	One metadata, all platforms
Deployment	Full application deployment	Metadata sync only

Who Should Use GTSuite?

- **Enterprise Software Teams:** Organizations needing rapid application development and frequent changes
- **Software Houses:** Companies building multiple applications for different clients
- **ISVs:** Independent software vendors requiring flexible, customizable solutions
- **Internal IT:** IT departments building line-of-business applications
- **Consultants:** Technical consultants implementing custom solutions quickly

2. The Metadata-Driven Philosophy

Separation of Concerns

GTSuite follows a strict separation between what the application does (metadata) and how it executes (framework code). The framework code remains stable and rarely changes, while metadata evolves continuously with business requirements.

Layer	Responsibility	Changes Frequency
Framework	Interpret and execute metadata	Rarely (bug fixes, new features)
Metadata	Define application behavior	Frequently (business changes)
Business Data	Store application records	Constantly (user operations)

Meta-Circular Design

A unique aspect of GTSuite is its meta-circular architecture: the designer tool itself is built using metadata principles. The Delphi designer application uses the same A1Suite components and metadata structures it generates for other applications. This "dogfooding" approach ensures the framework is robust and validates the architectural decisions.

> *"The tool that creates metadata is itself defined by metadata."*

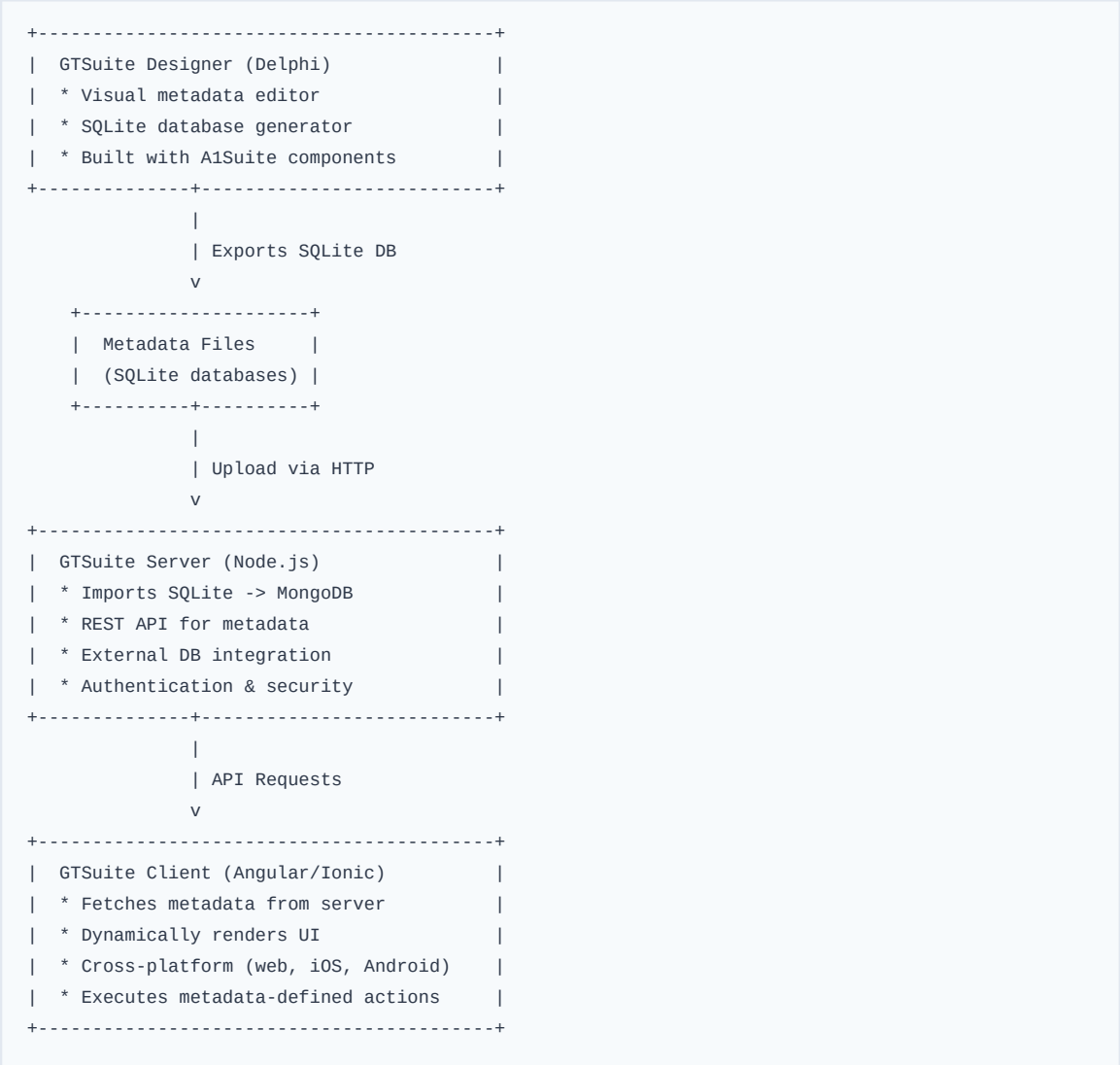
Configuration Over Code

- **Faster Development:** Visual design tools instead of manual coding
- **Easier Maintenance:** Changes in metadata, not scattered through code
- **Better Documentation:** Metadata itself documents the application
- **Consistency:** Enforced patterns and standards across all pages
- **Version Control:** Metadata tracked like code, with diff and merge
- **Testing:** Metadata validation catches errors before deployment

3. System Architecture

High-Level Overview

GTSuite consists of three main components working together in a metadata lifecycle:



The Metadata Journey

1. **Design:** Developer uses Delphi designer to create page layouts, forms, grids, menus
2. **Store:** Designer saves metadata to SQLite database file (e.g., GTS_GTSW.db)
3. **Export:** SQLite file is ready for deployment (version controlled, portable)
4. **Upload:** Built-in uploader transfers SQLite to Node.js server
5. **Import:** Server reads SQLite and imports into MongoDB collections
6. **Transform:** Data is normalized and optimized for runtime consumption
7. **Serve:** Server exposes REST API endpoints for metadata retrieval
8. **Request:** Angular/Ionic client requests page metadata when user navigates
9. **Render:** Client interprets metadata and dynamically generates UI
10. **Execute:** User interactions trigger actions defined in metadata
11. **Persist:** Data operations flow to external databases (Oracle, SQL Server, etc.)

4. The Three Pillars

Each component of GTSuite has a specific role in the metadata lifecycle. Together, they form a complete development and runtime environment.

4.1 GTSuite Designer (Delphi)

The Designer is a sophisticated Delphi application that serves as the visual development environment. It provides IDE-like tools for creating application metadata without writing code.

- **Visual Page Designer:** Drag-and-drop interface for forms and fields
- **Grid Designer:** Configure data grids with columns, sorting, filtering
- **Menu Designer:** Hierarchical menu structure with icons and permissions
- **SQL Repository:** Manage SQL queries and stored procedures
- **Action Builder:** Define user actions and business logic
- **Report Designer:** Create report templates and definitions
- **Connection Manager:** Configure database connections
- **Built-in Uploader:** Direct upload to Node.js server

Technology: Delphi, A1Suite Components, SQLite **Output:** SQLite database files (GTS_*.db)
Documentation: GTSuite_Designer_Documentation.pdf

4.2 GTSuite Server (Node.js)

The Server is the runtime backend that stores metadata in MongoDB, serves it via REST API, handles authentication, and proxies requests to external databases.

- **SQLite Importer:** Automated import from SQLite to MongoDB
- **Metadata API:** REST endpoints for page data, menus, translations
- **Authentication:** JWT tokens, OAuth (Google, Microsoft), 2FA
- **Database Proxy:** Execute queries on Oracle, SQL Server, PostgreSQL
- **File Management:** Upload, download, document handling
- **Email Services:** SMTP integration for notifications
- **Scheduled Tasks:** Cron job execution
- **Comprehensive Logging:** Auth, errors, queries, performance

Technology: Node.js, Express, MongoDB, Mongoose **Databases:** Oracle, SQL Server, PostgreSQL, SQLite **Documentation:** GTSuite_Server_Documentation.pdf

4.3 GTSuite Client (Angular/Ionic)

The Client is a hybrid mobile application that renders dynamic user interfaces based on metadata. Built with Angular and Ionic, it runs on web browsers, iOS, and Android from a single codebase.

- **Metadata Rendering:** Dynamically creates UI from server metadata
- **GTS Components:** Reusable form, grid, toolbar, tab components
- **Action Execution:** Interprets and executes metadata actions
- **Cross-Platform:** Web, iOS, Android from single codebase
- **DevExtreme UI:** Enterprise-grade data grids and components
- **Offline Support:** Local storage with Ionic Storage
- **Real-time Updates:** WebSocket integration
- **Standard Templates:** Consistent HTML structure across all pages

Technology: Angular 20, Ionic 8, TypeScript, Capacitor **Platforms:** Web, iOS, Android **Documentation:** GTSuite_Client_Documentation.pdf

5. The Development Workflow

Step-by-Step Process

1. Create Project: Copy GTS_Template.db → GTS_MYAPP.db

2. Open Designer: Launch GTS Designer, select MYAPP project

3. Configure: Set project ID, database connections, global settings

4. Design Menu: Create hierarchical menu structure

5. Create SQL: Define queries and stored procedures in SQL repository

6. Design Pages: Create forms, grids, fields using visual designer

7. Define Actions: Configure user actions and business logic

8. Test Locally: Validate metadata completeness

9. Upload: Use built-in uploader to transfer to Node.js server

10. Import: Server imports SQLite into MongoDB

11. Deploy Client: Client application fetches and renders metadata

12. Iterate: Make changes in designer, re-upload as needed

Iterative Development

GTSuite supports rapid, iterative development. Changes to existing pages can be made in the designer and uploaded without downtime. Users see changes immediately after refreshing their browser - no application redeployment needed.

6. Key Concepts

Projects

A project represents a complete application with its own metadata database. Each project has a unique ID (e.g., GTSW, GTR, DCW) and can have multiple pages, menus, and configurations.

Pages/Forms

Pages are the building blocks of the application. Each page has a unique URL, contains forms, grids, tabs, and toolbars, all defined in metadata.

Data Sets

Data sets connect UI components to database queries. They define what data to show and how to perform CRUD operations (Create, Read, Update, Delete).

Actions

Actions define user interactions: button clicks, menu selections, form submissions. Each action specifies what should happen (execute query, open form, show message, etc.).

Conditional Rules

Rules that control field visibility, enabled state, and validation based on data values. For example: "Show field B only when field A equals 'X'".

Templates

The GTSW project serves as the reference implementation. All projects follow the same patterns and HTML structure, ensuring consistency and reusability.

7. Benefits & Advantages

For Development Teams

- **Rapid Development:** Build applications 5-10x faster than traditional coding
- **Visual Tools:** Designer reduces coding to metadata configuration
- **Consistency:** Enforced patterns across all pages and projects
- **Reusability:** Share components, queries, and patterns across projects
- **Lower Skill Barrier:** Less programming expertise needed for UI design
- **Parallel Development:** Multiple developers work on different pages independently

For Business Stakeholders

- **Faster Time to Market:** Applications deployed in weeks instead of months
- **Lower Costs:** Less development time, easier maintenance
- **Flexibility:** Quick response to changing business requirements
- **Reduced Risk:** Changes don't require full application redeployment
- **Better Alignment:** Visual tools enable stakeholder participation in design
- **Scalability:** Add new features without architectural changes

For IT Operations

- **Simpler Deployments:** Metadata updates don't require downtime
- **Version Control:** SQLite files tracked in Git alongside code
- **Rollback:** Easy to revert to previous metadata versions
- **Testing:** Metadata validation before deployment
- **Monitoring:** Comprehensive logging of all operations
- **Multi-Environment:** Same code, different metadata per environment

8. Use Cases & Applications

Enterprise Resource Planning (ERP)

Create comprehensive ERP systems with modules for finance, inventory, HR, and sales. Each module as a separate project with shared authentication and infrastructure.

Customer Relationship Management (CRM)

Build CRM applications with custom fields, workflows, and integrations. Easily adapt to different industries and business processes.

Line of Business (LOB) Applications

Develop internal business applications for departments: project tracking, resource management, compliance systems, etc.

Data Entry & Management Systems

Create forms-heavy applications for data collection, validation, and management with complex validation rules.

Reporting & Analytics Dashboards

Build interactive dashboards with dynamic filters, drill-downs, and export capabilities connecting to multiple data sources.

Mobile Field Applications

Deploy cross-platform mobile apps for field workers: inspections, surveys, data collection with offline capabilities.

Real-World Example: GTSW

The GTSW (GTSuite Web) project is a complete administration module built with GTSuite that manages users, roles, permissions, database connections, logs, and system configuration. It serves as both a functional admin tool and a reference implementation demonstrating GTSuite's capabilities.

9. Technology Stack

Complete Technology Overview

Component	Technologies	Version
Designer	Delphi, SQLite, A1Suite	Delphi 11+
Server	Node.js, Express, MongoDB	Node 18+
Client	Angular, Ionic, TypeScript	Angular 20, Ionic 8
Databases	Oracle, SQL Server, PostgreSQL	Various
UI Framework	DevExtreme, Ionic Components	DevExtreme 25+
Mobile	Capacitor	7.4+
Security	JWT, OAuth 2.0, bcrypt	Latest

Why These Technologies?

- **Delphi:** Proven rapid application development platform with native Windows performance
- **SQLite:** Portable, self-contained database perfect for metadata transfer
- **Node.js:** High-performance JavaScript runtime for scalable backend
- **MongoDB:** Flexible NoSQL database ideal for storing nested metadata structures
- **Angular:** Mature, enterprise-grade frontend framework with strong TypeScript support
- **Ionic:** Leading hybrid mobile framework with native-like UI components
- **DevExtreme:** Enterprise-grade data grid and form components

10. Getting Started

Prerequisites & Licensing

GTSuite is provided with pre-compiled executables for the Designer and Report Server. The only commercial license required is DevExtreme for the Angular/Ionic client.

- **GTS Designer (Delphi):** Provided as compiled executable - no Delphi license needed
- **Report Server (FastReport):** Provided as compiled executable - FastReport included
- **Node.js Server:** Open source - no license required
- **Angular/Ionic Client:** Open source framework - no license required
- **DevExtreme:** COMMERCIAL LICENSE REQUIRED for production use
- **MongoDB:** Community Edition - free for most use cases
- **Databases:** Oracle, SQL Server, PostgreSQL - use existing licenses

IMPORTANT: DevExtreme is the only third-party commercial component requiring a license. It provides enterprise-grade data grids and UI components for the Angular/Ionic client. A DevExtreme license must be purchased from DevExpress for production deployments.

Quick Start Guide

1. **Install GTS Designer:** Run provided GTSDesigner.exe (no Delphi needed)
2. **Install Report Server:** Run provided GTSRptServer.exe (no FastReport needed)
3. **Set Up Server:** Install Node.js 18+, MongoDB, deploy GTSuite Server
4. **Set Up Client:** Install Angular CLI, Ionic CLI, npm install (including DevExtreme)
5. **Configure DevExtreme:** Add your DevExtreme license key to the client
6. **Create First Project:** Copy GTS_Template.db, rename to your project
7. **Design Simple Page:** Use GTS Designer to create a basic list page
8. **Upload Metadata:** Use designer's built-in uploader to push to server
9. **Run Client:** Start Angular dev server, navigate to your page
10. **Iterate:** Make changes, upload, refresh browser to see updates

Installation Checklist

Component	Source	License Required
GTS Designer	Provided compiled .exe	No - Included
Report Server	Provided compiled .exe	No - Included
Node.js	Download from nodejs.org	No - Open Source
MongoDB	Download from mongodb.com	No - Community Edition
Angular/Ionic	npm install	No - Open Source
DevExtreme	npm install devextreme	YES - Purchase from DevExpress

Learning Path

Week 1: Understand architecture, set up environment, explore GTSW reference project **Week 2:** Create simple CRUD pages, learn data sets and SQL binding **Week 3:** Master actions, conditional rules, and business logic **Week 4:** Advanced topics: reports, file uploads, API integration **Ongoing:** Build real projects, refer to documentation as needed

Resources

- **Documentation:** Five comprehensive PDF guides (intro + 3 technical + report server)
- **GTSW Reference:** Complete admin module as working example
- **Template Database:** GTS_Template.db as starting point
- **Sample Projects:** GTR, DCW as additional examples
- **DevExtreme License:** Purchase from https://js.devexpress.com/Buy/
- **Support:** Contact GTsoftware di Giancarlo Thiella for assistance

11. Documentation Guide

GTSuite documentation consists of four comprehensive PDF documents. Start with this introduction, then dive into the technical documentation for the components you're working with.

Document	When to Read	Focus Area
GTSuite_Introduction.pdf	START HERE - Overview	Architecture, concepts, workflow
GTSuite_Designer_Documentation.pdf	Creating metadata	Delphi designer, SQLite, metadata design
GTSuite_Server_Documentation.pdf	Server deployment	Node.js backend, API, database integration
GTSuite_Client_Documentation.pdf	Frontend development	Angular/Ionic, UI components, GTSW reference

Recommended Reading Order

For Architects/Team Leads: Read all four documents to understand complete system

For Metadata Designers: Introduction → Designer → Client (for understanding output)

For Backend Developers: Introduction → Server → Designer (to understand input)

For Frontend Developers: Introduction → Client → Designer (to understand metadata)

For New Team Members: Start with Introduction, then dive into your area

Conclusion

GTSuite represents a paradigm shift in application development - from code-centric to metadata-centric. By separating what an application does from how it executes, GTSuite enables unprecedented flexibility, maintainability, and development speed. The meta-circular design - where the designer tool itself is built with the same principles it enables - validates the architecture and provides a robust foundation for enterprise application development. Whether you're building internal LOB applications, customer-facing portals, or mobile field apps, GTSuite provides the tools and framework to deliver faster, with higher quality, and with the flexibility to evolve as requirements change. Welcome to the GTSuite ecosystem. We look forward to seeing what you'll build.

Contact & Support GTsoftware di Giancarlo Thiella For inquiries, support, or consulting services